

Implementasi OLAP pada Data Kerja Praktik dan Tugas Akhir Menggunakan *Framework Modular Cube JS*

I Komang Arya Ganda Wiguna*, Desak Putu Diah Kumala Dewi**, I Gede Iwan Sudipa***

* STMIK STIKOM Indonesia

** STMIK STIKOM Indonesia

*** STMIK STIKOM Indonesia

*kmaryagw@stiki-indonesia.ac.id, **desakdiah@stiki-indonesia.ac.id, ***iwansudipa@stiki-indonesia.ac.id

ABSTRACT

The government's role in the implementation of higher education in Indonesia is to provide a university database. It is hoped that the government and the public can participate in assessing and conducting surveillance with the database. Every tertiary institution must report everything related to the implementation of education, starting from lecturer data, student data, and lecture data. In its performance, a Dikti Feeder application has been prepared that can transmit data. Each university will adjust the data entry following the Dikti standards. As one of the higher education providers, STMIK STIKOM Indonesia has been able to report data well, but specifically for final assignments and work practices, it has not been maximized due to the development of a separate system from the academic system. For this reason, a suitable system will be developed to accommodate Thesis and Internship data related to reporting on the Dikti Feeder by applying the Online Analytical Processing (OLAP) method using the Cube JS modular framework. Some of the tests carried out are schema file testing, frontend and backend testing, cube client testing, querying testing and load request testing showing the data can be displayed correctly and the process is successful.

Keyword: OLAP, Cube JS, Data Analytic, Feeder Dikti

1. Introduction

Salah satu upaya pemerintah dalam menjamin pelaksanaan mutu pendidikan tinggi adalah dengan menyediakan pangkalan data pendidikan tinggi atau yang disebut dengan PDDikti. PDDikti merupakan kumpulan data penyelenggaraan kegiatan Pendidikan Tinggi seluruh Perguruan Tinggi yang terintegrasi secara nasional. Fungsi dari PDDikti tercantum dalam pasal 56 ayat 2 UU No. 12 Tahun 2012 yang menyebutkan PDDikti berfungsi sebagai sumber informasi bagi lembaga akreditasi untuk melakukan akreditasi, bagi pemerintah untuk melakukan pengawasan, pemantauan dan pembinaan, dan bagi masyarakat untuk mengetahui kinerja program studi dan Perguruan Tinggi[1].

STMIK STIKOM Indonesia (STIKI) sebagai salah satu perguruan tinggi swasta di Bali yang berfokus pada bidang teknologi informasi, maka seluruh data penyelenggaraan perkuliahan wajib dilaporkan secara benar. Jika tidak maka akan diberikan sanksi, sesuai peraturan Permenristekdikti No. 61 Tahun 2016 pasal 10 butir 7 dan pasal 12 butir 3[2]. Untuk mendukung proses pelaporan ke PDDikti, Kemenristekdikti dalam bidang Pangkalan Data Pendidikan Tinggi yang berada di bawah Pusadatin, sudah menyediakan sebuah *tools* pelaporan yang dapat digunakan oleh masing-masing perguruan tinggi. *Tools* tersebut adalah *Feeder* PDDikti yang dapat mengirimkan informasi terkait dengan penyelenggaraan kegiatan perguruan tinggi. Data pada *Feeder* akan terhubung dengan sistem Forum Laporan Pendidikan Tinggi (Forlap) dengan melakukan proses sinkronisasi data. Forlap sendiri merupakan sistem aplikasi *online report* secara keseluruhan yang dikelola oleh DIKTI. Data yang disediakan adalah data dosen, mahasiswa dan data perkuliahan[3][4].

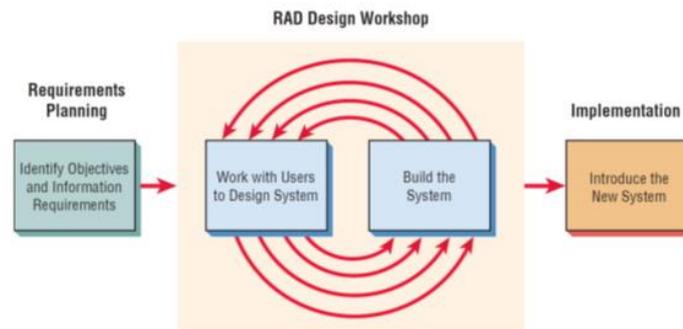
Dalam penggunaan aplikasi *Feeder* terdapat kendala pada saat proses input data dimana data akan dimasukkan satu per satu ke dalam aplikasi. Proses ini akan menghabiskan banyak waktu karena banyaknya data yang harus dilaporkan, untuk itu STIKI Indonesia menggunakan aplikasi tambahan yaitu *Feeder Importer* yang dapat melakukan import data melalui *microsoft excel* secara massal. Kompleksitas data serta proses pengolahan sangat mempengaruhi informasi yang dihasilkan[5][6]. Pada tiap semester data yang harus dilaporkan adalah data mahasiswa, data dosen, data kelas perkuliahan, data KRS mahasiswa, data ajar dosen, data Aktivitas Kuliah Mahasiswa (AKM), data nilai, data skripsi/kerja praktek dan lain-lainnya. Kendala yang terjadi adalah belum tersedianya kesesuaian data untuk pelaporan data Tugas Akhir (TA) dan Kerja Praktik (KP). Dalam pelaksanaannya proses TA dan KP sudah menggunakan sistem yang bernama *Sintesisys*[7], namun

belum disediakan fitur untuk melakukan *import data*. Data akan dianalisis sesuai dengan kebutuhan pada aplikasi *Feeder* dan disesuaikan dengan format pada aplikasi *Feeder Importer*.

Berdasarkan permasalahan tersebut maka dikembangkan sebuah web analitis dari data tugas akhir dan kerja praktek yang dilaporkan pada Pangkalan Data Dikti dengan menerapkan metode *Online Analytical Processing* (OLAP) dengan menggunakan *modular framework* Cube JS. Penggunaan *Cube JS* pada penelitian dikarenakan Cube JS bersifat *platform open source* yang mampu mengakses data melalui *Application Programming Interface* (API) serta menghasilkan analisis data menjadi lebih mudah dilengkapi dengan visualisasi data yang sesuai dengan kebutuhan pengguna[8] serta dapat diintegrasikan dengan konsep OLAP[9]. Tujuan pengembangan sistem yaitu membantu pihak terkait yaitu Bidang Pengelolaan Data Digital STIKI Indonesia dalam melakukan proses proses pelaporan data tugas akhir dan kerja praktek pada sistem *Sintesys*, sehingga dapat memudahkan operator dalam melakukan pelaporan serta proses pemeriksaan silang dapat dilakukan untuk mengecek validasi data. Beberapa penelitian yang menerapkan OLAP dalam data analitik khususnya dalam pengolahan data dosen dan karyawan[10], penelitian lainnya[11] menganalisis data kejadian bencana menggunakan metode OLAP serta penggunaan OLAP dalam analisis data pinjaman Bank[12].

2. Research Method

Metode penelitian yang digunakan pada penelitian ini adalah *Rapid Application Development* (RAD). Metode RAD digunakan karena tahap pengembangan sistem yang membutuhkan waktu yang relatif singkat[13]. Secara umum metode RAD dimulai dengan tahapan rencana kebutuhan (*requirement planning*), tahapan desain dan tahapan implementasi[13][14], dapat dilihat pada Gambar 1, berikut:



Gambar 1. Tahapan Metode *Rapid Application Development* (RAD) (Kendal, 2013)

2.1 Tahapan Rencana Kebutuhan (*Requirement Planning*)

Pada tahapan rencana kebutuhan dimulai dengan mendeskripsikan kebutuhan sistem. kegiatan yang dilakukan yaitu menganalisis kebutuhan sistem berdasarkan dari pengumpulan data dokumentasi. Dokumen diartikan sebagai suatu catatan tertulis atau gambar yang tersimpan tentang sesuatu yang sudah terjadi[15]. Kajian dokumen dilakukan dengan cara menganalisis data yang didapat dari aplikasi *Feeder Importer* untuk disesuaikan pada data aplikasi *Sintesys*. Pengumpulan data selanjutnya melakukan wawancara dengan bidang KP dan TA STMIK STIKOM Indonesia untuk memperoleh informasi dalam proses bisnis dan mekanisme pengumpulan file sistem. Tahapan rencana kebutuhan dilanjutkan dengan melakukan analisis sistem, yang dibagi menjadi 2 proses[16], yaitu:

1. Analisis sistem yang berjalan
Sistem yang dijalankan saat ini untuk melakukan proses yang terkait dengan kerja praktek dan tugas akhir masih belum mampu melakukan analisa secara menyeluruh dalam menghasilkan data yang mendukung proses pengecekan dan pelaporan data ke PDDikti. Sehingga akan menyulitkan admin sistem dalam melakukan proses pelaporan data.
2. Analisis sistem yang diusulkan
Berdasarkan gambaran dari sistem yang telah berjalan dapat diusulkan sebuah sistem analisis berbasis online yang dapat melakukan analisa data secara menyeluruh dalam mendukung proses pengecekan dan pelaporan data ke PDDikti. Sistem analisis data berbasis online yang diusulkan digambarkan dalam bentuk diagram umum sistem yang merupakan bagan yang menunjukkan alur kerja di dalam sistem secara keseluruhan dan menjelaskan prosedur – prosedur yang ada di dalam sistem.

2.2 Tahapan Desain

Tahapan selanjutnya adalah desain sistem yang nantinya akan menjadi acuan alur implementasi sistem informasi. Desain sistem dilakukan berdasarkan tahap kebutuhan sebelumnya yang diperoleh dari dokumentasi, wawancara serta analisis sistem berdasarkan kebutuhan user. Desain sistem yang digunakan

adalah dengan pendekatan berbasis obyek yaitu menggunakan usecase diagram. Usecase diagram berupa gambaran visual konteks yang menjelaskan interaksi antara pengguna dan sistem[17].

2.3 Tahapan Implementasi

Langkah selanjutnya setelah menganalisa kebutuhan sistem dan memetakan desain sistem adalah implementasi sistem. Tahapan ini bertujuan untuk menerapkan metode dan program sesuai dengan kebutuhan sistem. Kegiatan yang dilakukan adalah membangun sistem sesuai kebutuhan dan model yang sudah dibuat. Hasil yang didapatkan berupa sebuah web analitis dari data tugas akhir dan kerja praktek yang dilaporkan pada Pangkalan Data Dikti dengan menerapkan metode *Online Analytical Processing* (OLAP) dengan menggunakan *modular framework* Cube JS .

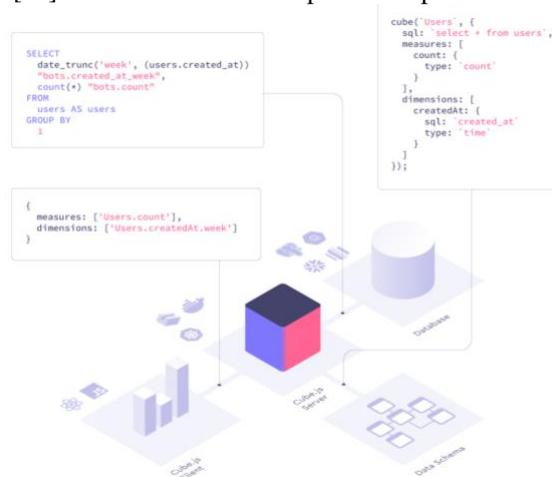
2.4 OLAP

OLAP adalah sebuah perangkat yang menggunakan teknologi visualisasi multidimensi sejumlah data untuk menyediakan akses yang lebih cepat bagi strategi informasi dengan tujuan mempercepat analisis. OLAP dapat mendukung dalam melakukan *query* pada *database* dengan cepat, mudah dan efisien serta mendukung *business intelligent*[18]. OLAP memiliki 5 teknik yang diringkas menjadi FASMI (*Fast Analysis of Shared Multi-dimensional Information*)[19], Teknik ini menjadi fitur unggulan dari OLAP yang digunakan dalam penelitian. Fitur OLAP dapat dijelaskan, sebagai berikut:

1. *Fast* yaitu sistem memiliki target untuk memberikan respon secepat mungkin pada pengguna menurut analisis yang dilakukan
2. *Analysis* yaitu sistem mampu mengatasi berbagai logika bisnis serta analisis statistik data yang relevan dengan aplikasi dan kebutuhan pengguna.
3. *Shared* yaitu melakukan keseluruhan keperluan pengamanan data, dimana kebutuhan dan kesediaan akses data oleh banyak pengguna disesuaikan dengan level pengguna.
4. *Multi-dimensional* yaitu sistem mampu memberikan representasi konseptual multidimensi data. Hal ini merupakan cara logis untuk melakukan analisis bisnis serta organisasi.
5. *Information* yaitu seluruh data serta informasi yang diperlukan dan relevan bagi aplikasi, mampu mengatasi adanya manipulasi data intuitif, fleksibilitas dalam menghasilkan laporan serta integrasi terhadap data warehouse.

2.5 Cube JS

Cube JS merupakan framework yang bersifat open source yang digunakan dalam membangun aplikasi web analitis. *Cube JS* digunakan dalam membangun aplikasi dengan *big data* dengan akselerasi query lebih cepat, keamanan dan control akses serta dapat mendesain *Application Programming Interface* (API)[20]. *Cube JS* mendukung pembuatan *dashboard* analisis yang dapat digunakan dalam *business intelligent*[18]. *Cube JS* bertindak sebagai *backend* analitik yang menerjemahkan logika bisnis ke dalam *SQL* dan melakukan penanganan pada koneksi basisdata. Komponen penting yang terkait pada *Cube JS* adalah *Cube JS Client*, *Cube JS Server* dan *data Schema*[21]. Arsitektur Cube JS dapat dilihat pada Gambar 2, berikut:



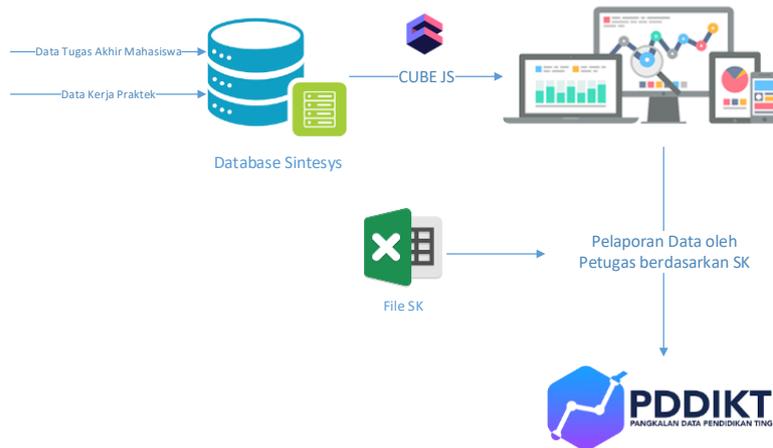
Gambar 2. Arsitektur Cube JS

Jika dilihat dari arsitektur, *Cube JS* bertindak sebagai backend analitik yang menerjemakan logika bisnis ke dalam *SQL* dan melakukan penanganan pada koneksi basisdata[22]. Komponen penting yang terkait pada *Cube JS* adalah *Cube JS Client*, *Cube JS Server* dan data *Schema*.

3. Result and Analysis

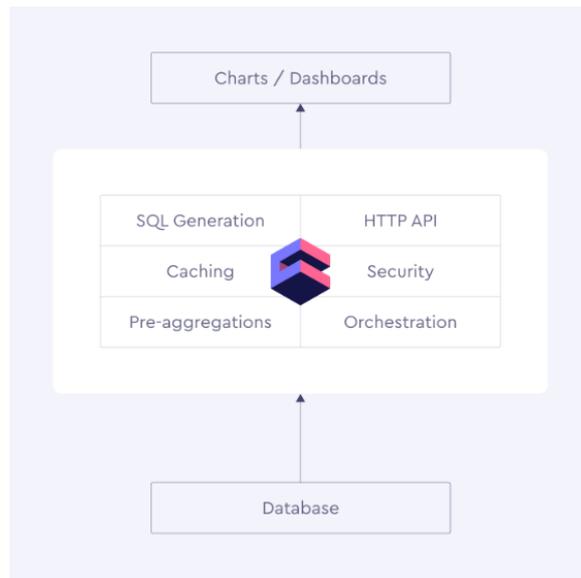
3.1. Gambaran Umum Sistem

Sistem yang akan dikembangkan diharapkan dapat memberikan informasi terkait data tugas akhir dan kerja praktek mahasiswa yang digunakan untuk bahan pelaporan pada aplikasi Feeder Dikti. Gambaran umum sistem ditunjukkan pada Gambar 3, berikut:



Gambar 3. Gambaran Umum Sistem

Pada gambaran umum sistem dapat dijelaskan pengembangan sistem dilakukan pada bagian output dimana akan menghasilkan sebuah file *excel* data tugas akhir dan kerja praktek. Format data akan mengikuti sesuai dengan format yang disediakan oleh aplikasi *Feeder Importer*. Sehingga pihak operator yang melakukan proses pelaporan langsung dapat melakukan input data secara massal. Arsitektur pada *Cube JS* digambarkan berikut ini.

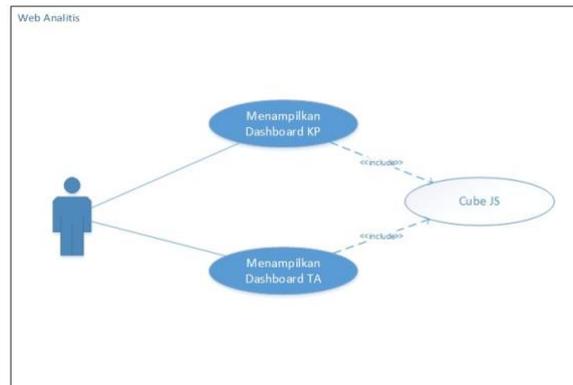


Gambar 3. Arsitektur Cube JS

3.2. Use Case Diagram

Perancangan sistem digunakan untuk memberikan gambaran yang lebih jelas mengenai sistem yang dibangun dan menjadi acuan untuk menghasilkan sistem yang sesuai. Penelitian ini menggunakan perancangan dengan model *Unified Modelling Language (UML)* dimana melibatkan beberapa pemodelan yaitu *use case*

diagram. *Use case diagram* merupakan diagram yang menggambarkan interaksi antara sistem dengan aktor siapa saja yang terlibat. Fungsionalitas dari sistem digambarkan dengan menggunakan *use case diagram*.



Gambar 2. Use Case Diagram

Pada gambar diatas menunjukkan *use case diagram* sistem web analitis, dimana terdapat 1 aktor yang terlibat yaitu user. User akan dapat melihat hasil analisa yang ditampilkan sistem yaitu dapat menampilkan *dashboard* data Kerja Praktik (KP) dan menampilkan *dashboard* data Tugas Akhir (TA).

3.3. Implementasi Cube JS

Dalam tahap implementasi *Cube JS* akan dibagi menjadi 2 proses utama yaitu :

- a) *Cube JS Backend*

Tahap ini akan membuat sebuah *Schema* yang digunakan sebagai *Object Relational Mapping* (ORM) untuk digunakan sebagai dasar dalam analitik dan memungkinkan untuk memodelkan berbagai macam kemungkinan.

```

cube(`Mahasiswa`, {
  sql: `SELECT * FROM akademik_sinte.kp_anggotas`,
  joins: {
    KpPengajuan: {
      relationship: `hasOne`,
      sql: `${Mahasiswa}.ang_kp_kode = ${KpPengajuan}.pkp_kode`
    }
  },
  measures: {
    count: {
      type: `count`,
      drillMembers: [createdAt, updatedAt]
    },
    mahasiswaLulus: {
      type: `count`,
      filters: [
        { sql: `${CUBE}.ang_nilHuruf IN ('A','B','C')` }
      ]
    }
  },
  dimensions: {
    id: {
      sql: `ang_id`,
      type: `number`,
      primaryKey: true
    },
    angKpKode: {
      sql: `ang_kp_kode`,
      type: `string`
    },
  },
}

```

```

angNim: {
  sql: `ang_nim`,
  type: `string`
},
createdAt: {
  sql: `created_at`,
  type: `time`
}
}
});
    
```

Gambar 3. Source Code Backend Cube JS

b) *Cube JS Frontend*

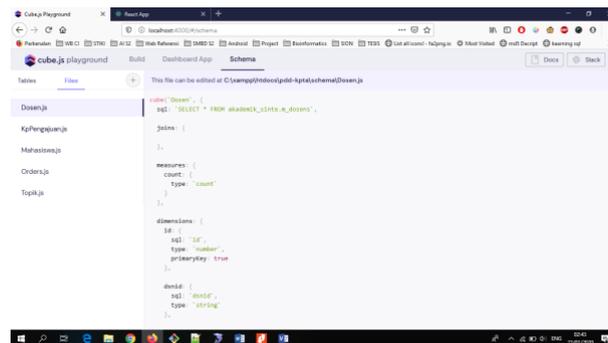
Pada tahap ini akan dibuat 2 buah file JS yaitu *Cube JS Javascript Client* untuk melakukan proses *query* dan *Cube JS React* yang digunakan sebagai modular framework dalam pengembangam sistem.

3.4. Implementasi Antarmuka

Hasil dari tampilan antarmuka sistem digambarkan dalam dua kelompok yaitu untuk *backend* dan *frontend*. Kedua implementasi ini digunakan sebagai penghubung user dalam melakukan interaksi dengan sistem.

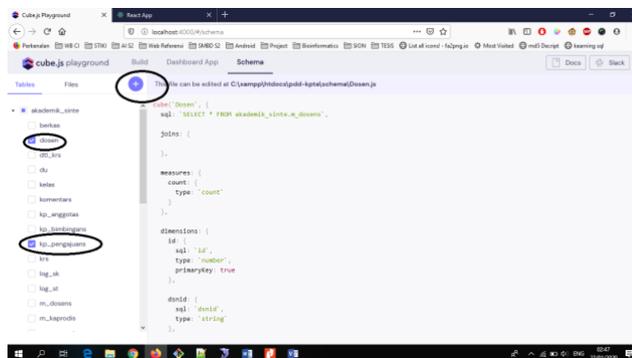
3.4.1. Antarmuka Backend

Tampilan pertama adalah untuk membuat file *Schema*. File ini bisa dibuat dengan cara manual dengan menambahkan file JS pada folder schema.



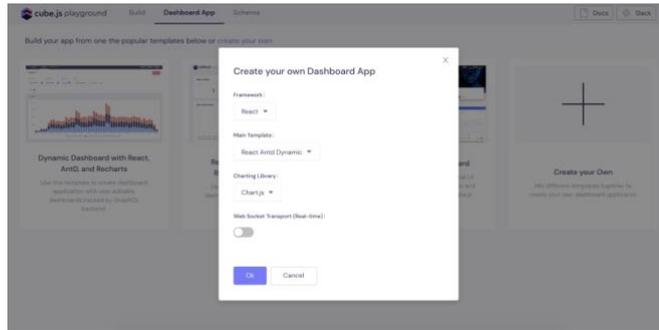
Gambar 4. Tampilan Backend Menu Schema

Jika user ingin membuat sebuah schema dari table database user dapat memilih tab Tables pada menu Schema. Pilih table apa yang ingin dibuatkan schema dengan cara mencentang kemudian tekan tombol tambah (+).



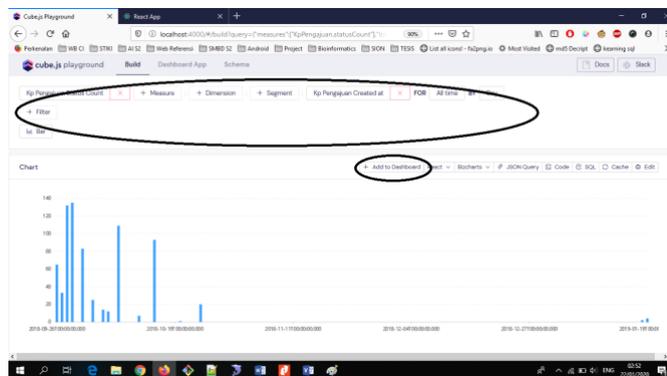
Gambar 5. Tampilan Backend Membuat Schema

Untuk menambahkan chart/grafik pada dashboard silahkan masuk pada menu Build. Tapi sebelum itu pastikan untuk memilih template yang bisa dipilih mana menu Dashboard App. Pilih sesuai dengan yang diinginkan kemudian pilih create lalu tunggu hingga proses selesai.



Gambar 6. Tampilan Membuat *Dashboard*

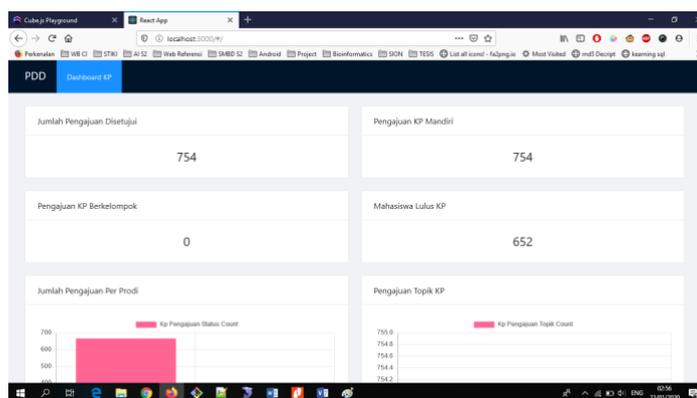
Untuk proses penambahan chart dapat dilihat pada gambar dibawah ini. Tipe dari grafik yang akan dibuat bisa disesuaikan karena sudah disediakan beberapa tampilan penyajian data dari line, bar, pie atau number. Jika proses konfigurasi sudah selesai pastikan untuk menekan tombol Add to Dashboard.



Gambar 7. Tampilan Membuat Grafik Sesuai dengan Schema

3.4.2. Antarmuka *Frontend*

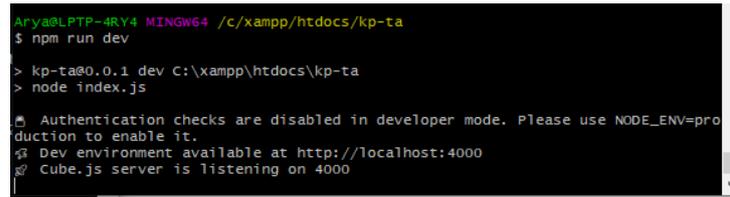
Halaman frontend ini akan diakses oleh pengguna yang terkait. Data sudah ditampilkan dalam bentuk grafik-grafik sehingga user dapat melakukan proses pemeriksaan silang. Informasi yang ditampilkan yaitu ada jumlah pengajuan kerja praktek yang disetujui, jumlah pengajuan kerja praktek yang mandiri, jumlah pengajuan kerja praktek berkelompok, rasio pengajuan kerja praktek per prodi, trend dari jumlah topik yang diajukan dan informasi lainnya.



Gambar 8. Tampilan Dashboard Web

3.5. Pengujian

Pengujian pada penelitian ini dilakukan untuk menguji aplikasi web analitik yang diimplementasikan dengan menggunakan framework Cube JS dipastikan dapat berjalan dengan baik. Tahap pengujian dimulai dari menjalankan aplikasi dengan menggunakan perintah pada CMD.



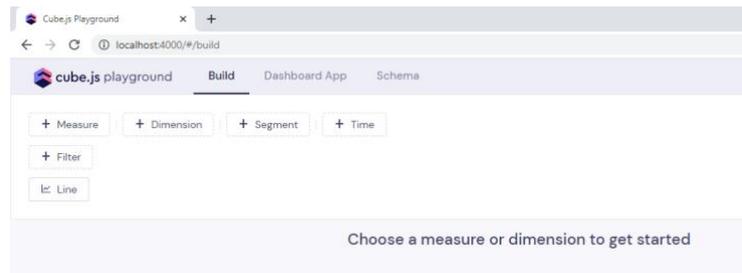
```
Arya@LPTP-4RY4 MINGW64 /c:/xampp/htdocs/kp-ta
$ npm run dev

> kp-ta@0.0.1 dev C:\xampp\htdocs\kp-ta
> node index.js

Authentication checks are disabled in developer mode. Please use NODE_ENV=production to enable it.
Dev environment available at http://localhost:4000
Cube.js server is listening on 4000
```

Gambar 9. Pengujian Menjalankan Cube JS

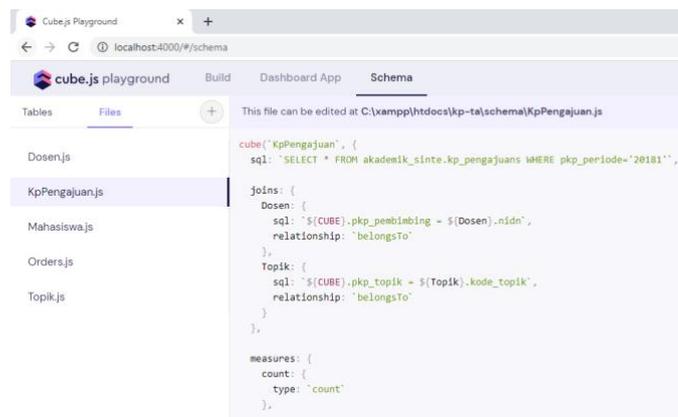
Untuk menjalankan aplikasi buka web browser kemudian masukkan alamat <http://localhost:4000>. User akan diarahkan pada halaman Build untuk membuat sebuah grafik pengukuran.



Gambar 10. Pengujian Halaman Membuat Grafik

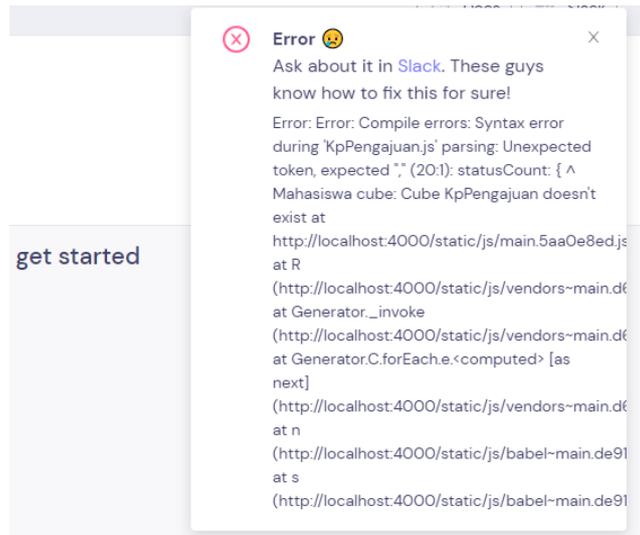
3.5.1 Pengujian File Schema

Selanjutnya adalah pengujian *file Schema* dimana pada tahap ini user membuat sebuah file dan disimpan dalam folder *Schema*.



Gambar 11. Pengujian Membuat File Schema

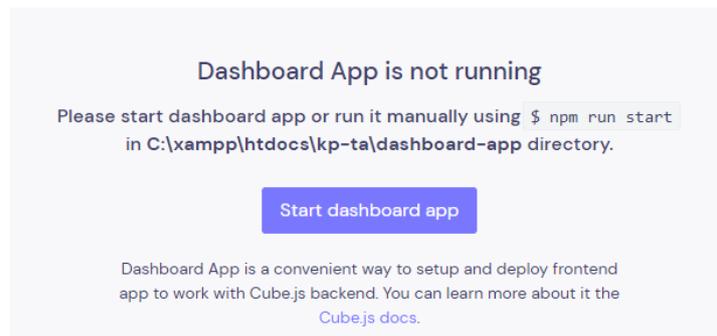
Jika pada *schema* terdapat *error* maka akan muncul tampilan *error* pada halaman *build* seperti berikut ini.



Gambar 12. Pengujian Error Saat Membuat Grafik

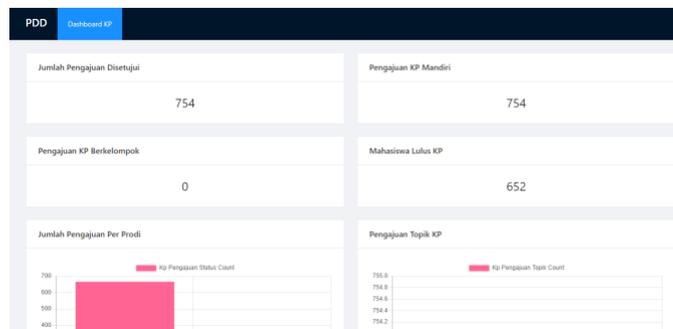
3.5.2 Pengujian Tampilan Frontend

Untuk mengakses halaman *frontend* pastikan untuk menjalankan server Dashboard App terlebih dahulu.



Gambar 13. Pengujian Saat Aplikasi belum dijalankan

Server *Dashboard App* akan dilanjutkan pada alamat <http://localhost:3000>. Ketika proses menampilkan akan dilakukan proses *query* pada *database* sesuai dengan *file schema*. Tampilan pada dashboard akan ditentukan berdasarkan parameter *measure* atau *dimension*.



Gambar 14. Pengujian Halaman *Dashboard*

3.5.3 Pengujian Cube Client

Pada *framework Cube JS*, *Cube JS Client* akan melakukan proses query berdasarkan *dimension*, *measure* dan *filter* yang telah didefinisikan. Selanjutnya *server* menggunakan *Schema Cube JS* untuk melakukan *generate SQL*. Hasil yang diperoleh kemudian dikirimkan kembali pada *Client* untuk ditampilkan dalam bentuk visualisasi. Pengujian dilakukan dengan melakukan pengecekan *request* data, misalkan pada bagian untuk mencari jumlah pengajuan KP yang disetujui. Hasil menjalankan *query* sebagai berikut.

```

Performing query:
SELECT
  count(CASE WHEN (`kp_pengajuan`.pkp_status = 1) THEN `kp_pengajuan`.id
END) `kp_pengajuan__status_count`
FROM
  (SELECT * FROM akademik_sinte.kp_pengajuan WHERE pkp_periode='20181')
AS `kp_pengajuan`
LIMIT 10000
{
  "queueSize": 5,
  "queuePrefix": "SQL_QUERY_STANDALONE_default",
  "requestId": "a965a80bd69298d4-1"
}
Performing query completed: (20ms)
SELECT
  count(CASE WHEN (`kp_pengajuan`.pkp_status = 1) THEN `kp_pengajuan`.id
END) `kp_pengajuan__status_count`
FROM
  (SELECT * FROM akademik_sinte.kp_pengajuan WHERE pkp_periode='20181')
AS `kp_pengajuan`
LIMIT 10000
{
  "queueSize": 5,
  "queuePrefix": "SQL_QUERY_STANDALONE_default",
  "requestId": "a965a80bd69298d4-1"
}

```

Gambar 15. Pengujian *Performing Query*

Dari sisi client akan dipanggil sebuah *request* terlebih dahulu, baru dijalankan proses *query*. Berikut hasil *request* yang sukses.

```

Load Request Success: {
  "query": {
    "measures": [
      "KpPengajuan.statusCount"
    ],
    "timeDimensions": [
      {
        "dimension": "KpPengajuan.createdAt"
      }
    ],
    "filters": []
  },
  "authInfo": {
    "iat": 1579614147,
    "exp": 1579700547
  },
  "requestId": "a965a80bd69298d4-1"
}

```

Gambar 14. Pengujian *Load Request*

Proses menampilkan grafik dengan menggunakan *react* dan berdasarkan dari parameter grafik yang didefinisikan. Hasil tampilan grafik sukses akan seperti berikut.



Gambar 15. Tampilan Data Hasil *Request*

4. Conclusion

Beberapa kesimpulan yang diperoleh adalah sebagai berikut :

1. Hasil penelitian yang dilakukan menunjukkan web analitis berhasil menampilkan rangkuman data kerja praktik dan tugas akhir dimana output yang ditampilkan dalam bentuk grafik, sehingga diharapkan dapat membantu pemeriksaan data dalam proses pelaporan data PDDikti. Proses *import* data yang belum disediakan pada sistem sebelumnya dapat diatasi pada penelitian ini dengan adanya proses tampilan hasil grafik berdasarkan dari *file schema* yang terdapat *query* untuk memperoleh data sesuai kebutuhan pengguna.
2. Sistem berhasil diimplementasi dengan menggunakan metode Online Analytical Processing (OLAP) yang dibangun di atas modular framework Cube JS dan React.
3. Hasil analisis data dengan *Cube JS* lebih baik dari analisis data yang sebelumnya digunakan dan dibuktikan dengan proses pengujian yaitu dari hasil implementasi *Cube JS* Backend diperoleh hasil bahwa *Cube JS* mampu digunakan dalam membuat *Schema* yang digunakan sebagai ORM dalam dasar analitik data KP dan TA. Dari implementasi *Cube JS Front end* terdapat hasil informasi dalam grafik jumlah pengajuan kerja praktek yang disetujui, jumlah pengajuan kerja praktek yang mandiri, jumlah pengajuan kerja praktek berkelompok, rasio pengajuan kerja praktek per prodi, trend dari jumlah topik yang diajukan dan informasi lainnya. Dari tahap pengujian untuk menampilkan hasil grafik sudah berhasil dilakukan, pengujian *file schema* dan tampilan front end sudah berhasil menampilkan *Server Dashboard App* yang digunakan untuk menampilkan akan dilakukan proses *query* pada *database* sesuai dengan *file schema*. Tahap Pengujian *cube client* yang terdiri dari pengujian *performing query* dan *load request* diperoleh hasil bahwa terdapat 754 data request yang berhasil ditampilkan dan sesuai dengan data master.

Beberapa hal yang di sarankan untuk pengembangan lebih lanjut:

1. Sebagai saran untuk pengembangan selanjutnya untuk lebih memperluas kategori data.
2. Penambahan web Security agar keamanan data lebih terjamin.

References

- [1] S. Widodo, H. Brawijaya, S. Samudi, and E. Retnoningsih, "Integrasi Data Akademik Dengan Aplikasi Feeder PDDIKTI Berbasis Web service," *BINA Insa. ICT J.*, vol. 5, no. 2, pp. 153–162, 2018.
- [2] A. C. Nugraha and N. Syakrani, "Aplikasi Audit Mutu Internal Online Studi Kasus SPM Politeknik Negeri Bandung," *J. DIFUSI*, vol. 1, no. 2, 2018.
- [3] V. H. Pranatawijaya, "Implementasi Pencatatan Aktivitas Mahasiswa Menggunakan Web Service Pada Feeder Pddikti Dengan Metode Extreme Programming," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 14, no. 2, pp. 179–188, 2020.
- [4] A. Siswanto and A. Rahim, "APLIKASI SINKRONISASI DATABASE ANTARA SIAK STIKOM DINAMIKA BANGSA JAMBI DENGAN PdDikti," *J. Process.*, vol. 12, no. 1, pp. 974–986, 2018.
- [5] N. L. W. S. R. Ginantra *et al.*, *Basis Data: Teori dan Perancangan*. Yayasan Kita Menulis, 2020.
- [6] I. G. I. Sudipa and E. A. P. Lestari, "RANCANG BANGUN SISTEM INFORMASI PENDUDUK DUSUN (STUDI KASUS: DUSUN TEGAL KORI KAJA UBUNG)," *J. Teknol. Inf. dan Komput.*, vol. 5, no. 2, 2019.
- [7] W. G. S. Parwita and N. W. S. Saraswati, "Pengembangan Sistem Monitoring Tugas Akhir dan Kerja Praktek STMIK STIKOM Indonesia," *S@ CIES*, vol. 7, no. 1, pp. 47–53, 2016.
- [8] D. Szelogowski, "Optimizing Data Cube Visualization for Web Applications: Performance and User-Friendly Data Aggregation," *arXiv Prepr. arXiv2101.00171*, 2021.
- [9] T. Hsiao, W.-S. Luk, and S. Petchulat, "Data visualization on web-based OLAP," in *Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP*, 2011, pp. 75–82.
- [10] A. Cucus and R. ROSITA, "Implementasi OLAP Untuk Efektivitas Pelaporan Data (Study Kasus Data Dosen

- dan Karyawan),” *Explor. J. Sist. Inf. dan Telemat. (Telekomunikasi, Multimed. dan Inform.,* vol. 8, no. 1, 2017.
- [11] R. Darman, “Analisis data kejadian bencana angin puting beliung dengan metode online analytical processing (OLAP),” *SINTECH (Science Inf. Technol. J.,* vol. 2, no. 1, pp. 18–23, 2019.
- [12] A. H. Widaningrum, “Analisis Data Peminjaman Bank Menggunakan Metode OLAP,” *J. Inform. Upgris,* vol. 4, no. 1, pp. 117–119, 2018.
- [13] K. E. Kendall and J. E. Kendall, “Rapid Application Development,” *Syst. Anal. Des.,* pp. 163–165, 2013.
- [14] B. Arinze and O. Amobi, “A Methodology for Developing Business Intelligence Systems,” in *Business Intelligence Techniques,* Springer, 2004, pp. 181–195.
- [15] B. Prsetyo and L. M. Jannah, “Metode penelitian kuantitatif,” 2019.
- [16] I. K. A. G. Wiguna and K. N. Semadi, “SISTEM INFORMASI MANAJEMEN PENGELOLAAN DATA BENCANA BERBASIS WEB DI BPBD KOTA DENPASAR,” *J. Teknol. Inf. dan Komput.,* vol. 5, no. 2, 2019.
- [17] T. A. Kurniawan, “Pemodelan use case (UML): evaluasi terhadap beberapa kesalahan dalam praktik,” *J. Teknol. Inf. dan Ilmu Komput,* vol. 5, no. 1, p. 77, 2018.
- [18] I. Tresnawati and E. Susilowati, “Implementasi Teknologi OLAP Pada Sistem Pengolahan Data Penjualan,” *Pros. Semnastek,* vol. 1, no. 1, 2014.
- [19] G. GHUFRON, B. Surarso, and R. Gernowo, “SISTEM INFORMASI CLUSTERING KUALITAS PROGRAM STUDI MENGGUNAKAN OLAP (ONLINE ANALYTICAL PROCESSING) DAN K-MEDOIDS.” School of Postgraduate, 2020.
- [20] N. Bevacqua, *Mastering Modular JavaScript.* “O’Reilly Media, Inc.,” 2018.
- [21] G. Fink and I. Flatow, “Modular JavaScript Development,” in *Pro Single Page Application Development,* Springer, 2014, pp. 35–48.
- [22] G. Fernandes, F. Portela, and M. F. Santos, “Towards the development of a Data Science Modular Solution,” in *2019 7th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW),* 2019, pp. 96–103.